Lecture 3 R Basics: Generate random variable

Ziwen Ye

Stevens Institute of Technology

zye2@stevens.edu

September 14, 2018

Agenda

1. Coin Flips
   - Sample()
   - Plot()
2. Generating Random Numbers
3. Binomial Distribution
4. Negative binomial distribution

Coin Flips

Before we go to generate random numbers lets make a function.

Suppose you are flipping a fair coins 1000 times. Simulate the probability of heads after each flip. then make a 2-D graph for that probability. On your graph, x should be the number of flips and y should be the probability of heads. We are expecting the curve converges to $1/2$ since the coin is fair.

Coin Flips

Analysis:

- We would keep generating a logical variable, using 1 for heads and 0 for tails.
- We also need a vector, which has 1000 elements, recording how many heads we got after each iteration.
- Another vector can be used for recording 1000 probabilities.
- Functions will be used: sample(), plot()

Sample()

sample takes a sample of the specified size from the elements of x using either *with* or *without* replacement.

- sample(x, size, replace = FALSE, prob = NULL)

Example

```
# take samples from population with replacement
> sample(x=c(1,2,3),2,replace = T)
[1] 1 1

# take samples from population without replacement
> sample(x=c(1,2,3),2,replace = F)
[1] 1 3
```

Plot()

Generic function for plotting of R objects. For more details about the graphical parameter arguments.

- plot(x, y, type, col, main)
- x,y are the coordinates of points in the plot
- type will determine plot type. "p" for points, "l" for lines.
- col will determine the color for points or lines
- main will determine the title for this plot

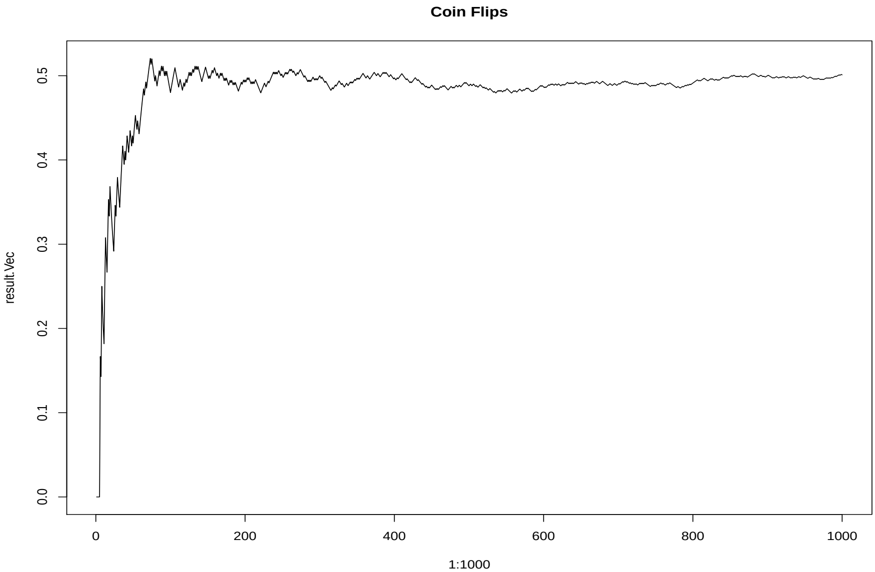
In future we will talk about plot() and GGplot() in detial.

Coin Flips

Example

```
> No.heads <- 0
> result.Vec <- NULL
> for (flips in 1:1000)
+ {
+     # x = c(1, 0), 1 means head, 0 means tail
+     tmp <- sample(x=c(1, 0), size=1, replace=T,
+                   prob=c(0.5, 0.5))
+     # add tmp to number of head
+     No.heads <- No.heads + tmp
+     result.Vec <- c(result.Vec,No.heads/flips)
+ }
> # produce a figure
> plot(1:1000, result.Vec, type="l", main=c("Coin Flips"))
```

Coin Flips



Coin Flips
0.0
0.1
0.2
0.3
0.4
0.5
result.Vec
0
200
400
600
800
1000
1:1000

Coin Flip Function

Example

```
coinFlip <- function(headProb) {
    No.heads <- 0
    result.Vec <- NULL
    for (flips in 1:1000)
   {
      tmp <- sample(x=c(1, 0), size=1, replace=T,
      prob=c(headProb, 1-headProb))
      No.heads <- No.heads + tmp
      result.Vec <- c(result.Vec,No.heads/flips)
   }
plot(1:1000, result.Vec, type="l")
}
```

Coin Flip Function

Now we have defined a function, to call it in the correct way you need to pass parameters with right type. In this case it has to be a real number between 0 and 1.

Example

coinFlip(0.5)
coinFlip(0.7)
coinFlip(0.9)
coinFlip(1)

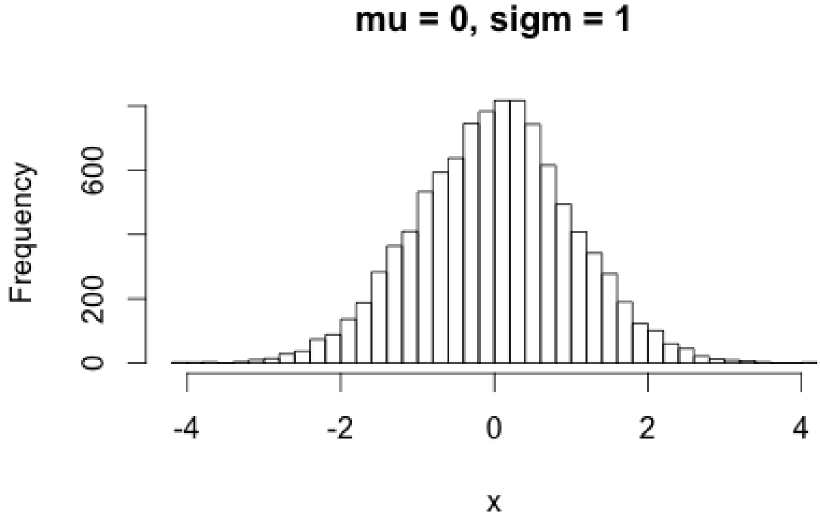
Normal Distribution

Now let's move on how to generate random variables.

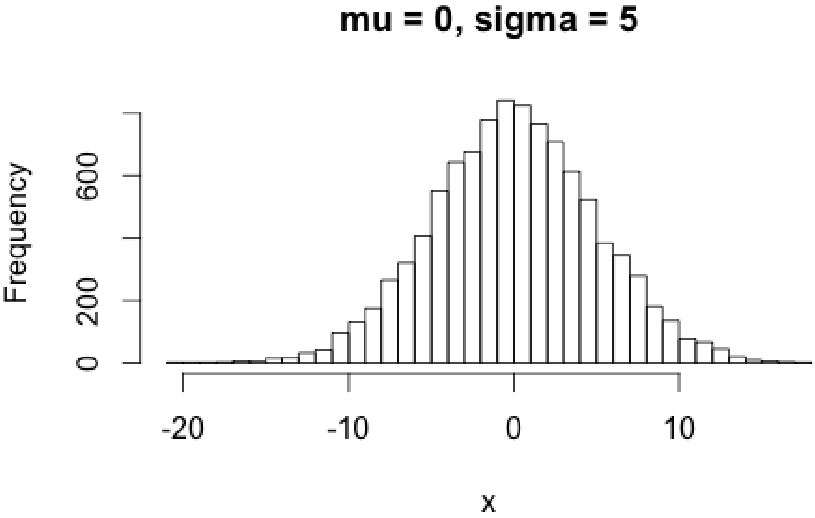
Example

```
> # rnorm(n = , mean = , sd = )

> x <- rnorm(n = 10000, mean = 0, sd = 1)
> hist(x)
> hist(x, nclass = 40)

> # another sigma
> x <- rnorm(n = 10000, mean = 0, sd = 5)
```

Normal Distribution



Normal Distribution



Normal Distribution

set.seed()

Anywho the random numbers R gives you aren't really random. They're pseudo-random. Basically there's a function that outputs numbers that look random. To do this it needs some inputs. The first input it gets will be the 'seed'.

Example

```
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
> rnorm(5)
[1] -0.8204684  0.4874291  0.7383247  0.5757814 -0.3053884
> set.seed(1)
> rnorm(5)
[1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078
```

Normal Distribution

- rnorm: generate random Normal variates with a given mean and standard deviation.
- dnorm: evaluate the Normal probability density (with a given mean/SD) at a point (or vector of points)
- pnorm: evaluate the cumulative distribution function for a Normal distribution
- qnorm: gives the quantile function

Normal Distribution

Example

```
> # Density function
> dnorm(x = 0)     # mean = 0, sd = 1
[1] 0.3989423
> dnorm(x = 1)     # check table if you want
[1] 0.2419707

> # cumulative distribution function
> pnorm(q = 0)
[1] 0.5
> pnorm(q = 5)
[1] 0.9999997
```

Other Distributions

- rt() – t distribution
- rpois() – poisson distribution
- runif() – uniform distribution
- rexp() – exponential distribution

Other Distributions

Example

```
> x <- rpois(1000, lambda = 2)
> hist(x, nclass = 40)
>
> x <- rexp(1000)
> hist(x, nclass = 40)
>
> x <- rt(1000, df = 10)
> hist(x, nclass = 40)
```

Binomial distribution

- In probability theory and statistics, the binomial distribution $\mathbf{B}(n, p)$ is a discrete probability distribution
- n stands for number of trials, p stands for probability to observe a success

Example

Assuming we are tossing a fair coin, what's the probability to observe 2 heads after 5 trials?

Analysis

The answer for the example should be

$$\mathbf{Pr}(\mathbf{X} = 2) = \binom{5}{2} 0.5^3 0.5^2$$

- Only two possible outcomes when tossing a coin: head and tail.
- The sequence to observe 2 heads is **not** important.

Negative binomial distribution

- Negative binomial is an extension of Geometric distribution.
- Detailed definition for this distribution can be different among text books. In this class, we will use the version which is used by R.
- $$Pr(x = r) = \binom{n + r - 1}{r} p^n (1 - p)^r \tag{1}$$

Negative binomial distribution

- In probability theory and statistics, the negative binomial distribution $\mathbf{NB}(r;p)$ is a discrete probability distribution.
- r stands for number of **failure**, p stands for probability to observe a success.
- **The last observation is a success.**

Example

Assuming we are tossing a fair coin, what's the probability to observe **the 2nd** head at **the 5th** trial?

Comparison between $\mathbf{B}(n, p)$ and $\mathbf{NB}(r; p)$

- In negative binomial distribution, the last observation is fixed (success).
- If you draw a random number from negative binomial distribution, the number of observation can be positive infinity.
- In negative binomial distribution, you can analyze this question by counting the number of failures (Default setting in R)

Analyze: Count the number of failures

- In total, we need two heads observation. n = 2
- In total, we need three tails observation. r = 3
- At the 5th toss, the observation has to be a head

$$\mathbf{Pr}(\mathbf{X} = k) = \mathbf{Pr}(\mathbf{X} = 3) = \binom{4}{3} 0.5^3 0.5^1 * \mathbf{0.5}$$